Tri_Group_inflation_comparison

February 12, 2025

1 Setup and Data

[1]:

```
from inflation_analysis import calculate_price_indexes, tri_grouping,␣
 ↪output_data, output_obs_table, price_index_over_time,␣
 ↪top_abs_weight_differences, top_price_index_contributors
```

[2]:

```
# Parameters
start_year = 2021
end_year = 2022
data_folder="/Users/roykisluk/Downloads/Consumer_Expenditure_Survey/"
top_n = 10
base_year = start_year
comparison_year = end_year

# Grouping
demo, income, ses, total_mmb = tri_grouping(start_year, end_year,␣
 ↪cex_data_folder = data_folder)
```

[ ]:

```
# Prepare data: calculate price indexes for each group, secondary and primary␣
 ↪categories, and total
demo_analysis, demo_mmb = output_data(demo, start_year, end_year, base_year,␣
 ↪top_n, data_folder)
income_analysis, income_mmb = output_data(income, start_year, end_year,␣
 ↪base_year, top_n, data_folder)
ses_analysis, ses_mmb = output_data(ses, start_year, end_year, base_year,␣
 ↪top_n, data_folder)

# General population
print("Calculating price indexes for general population...")
gen_pop_df, gen_pop_secondary_df, gen_pop_primary_df,␣
 ↪gen_pop_yearly_price_index = calculate_price_indexes(start_year, end_year,␣
 ↪base_year, cex_data_folder=data_folder, verbose=False)
gen_pop = {
    'combined_secondary_df': gen_pop_secondary_df,
    'combined_primary_df': gen_pop_primary_df,
    'yearly_price_index': gen_pop_yearly_price_index
}
```

```python
print("Done.")
```

```
Group 1/4 (Arabs) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.24it/s]
Calculating price indexes: 100%|      | 2/2 [00:02<00:00,  1.24s/it]

Group 1/4 (Arabs) successfully computed.
Group 2/4 (Haredi) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.45it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.29it/s]

Group 2/4 (Haredi) successfully computed.
Group 3/4 (Young) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.93it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.09it/s]

Group 3/4 (Young) successfully computed.
Group 4/4 (Old) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.71it/s]
Calculating price indexes: 100%|      | 2/2 [00:02<00:00,  1.19s/it]

Group 4/4 (Old) successfully computed.
Group 1/10 (1) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.85it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.32it/s]

Group 1/10 (1) successfully computed.
Group 2/10 (2) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.86it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.34it/s]

Group 2/10 (2) successfully computed.
Group 3/10 (3) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.69it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.30it/s]

Group 3/10 (3) successfully computed.
Group 4/10 (4) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.95it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.09it/s]

Group 4/10 (4) successfully computed.
Group 5/10 (5) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.97it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.28it/s]

Group 5/10 (5) successfully computed.
Group 6/10 (6) started.
```

```
Loading price data: 100%|      | 2/2 [00:01<00:00,  1.93it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.22it/s]

Group 6/10 (6) successfully computed.
Group 7/10 (7) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.97it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.24it/s]

Group 7/10 (7) successfully computed.
Group 8/10 (8) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.97it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.21it/s]

Group 8/10 (8) successfully computed.
Group 9/10 (9) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.97it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.15it/s]

Group 9/10 (9) successfully computed.
Group 10/10 (10) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.95it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.12it/s]

Group 10/10 (10) successfully computed.
Group 1/5 (1) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.97it/s]
Calculating price indexes: 100%|      | 2/2 [00:02<00:00,  1.05s/it]

Group 1/5 (1) successfully computed.
Group 2/5 (2) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.84it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.13it/s]

Group 2/5 (2) successfully computed.
Group 3/5 (3) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.38it/s]
Calculating price indexes: 100%|      | 2/2 [00:02<00:00,  1.13s/it]

Group 3/5 (3) successfully computed.
Group 4/5 (4) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.97it/s]
Calculating price indexes: 100%|      | 2/2 [00:03<00:00,  1.83s/it]

Group 4/5 (4) successfully computed.
Group 5/5 (5) started.

Loading price data: 100%|      | 2/2 [00:01<00:00,  1.59it/s]
Calculating price indexes: 100%|      | 2/2 [00:01<00:00,  1.74it/s]
```

```
Group 5/5 (5) successfully computed.
```

```
Loading price data: 100%|      | 2/2 [00:01<00:00,  1.92it/s]
Calculating price indexes: 100%|      | 2/2 [00:05<00:00,  2.92s/it]
```

2 Output

2.1 Tables

[4]:
```
# Observations tables
output_obs_table(start_year, end_year, demo_mmb)
```

```
+--------+---------------+---------------+
|        | 2021          | 2022          |
|--------+---------------+---------------|
| Arabs  | 951 (15.79%)  | 727 (13.31%)  |
| Haredi | 551 (9.15%)   | 595 (10.89%)  |
| Young  | 877 (14.56%)  | 820 (15.01%)  |
| Old    | 1779 (29.53%) | 1663 (30.44%) |
| Total  | 6024 (100.0%) | 5463 (100.0%) |
+--------+---------------+---------------+
```

[5]:
```
output_obs_table(start_year, end_year, income_mmb)
```

```
+-------+---------------+---------------+
|       | 2021          | 2022          |
|-------+---------------+---------------|
| 1     | 503 (8.35%)   | 515 (9.43%)   |
| 2     | 517 (8.58%)   | 537 (9.83%)   |
| 3     | 548 (9.1%)    | 554 (10.14%)  |
| 4     | 563 (9.35%)   | 562 (10.29%)  |
| 5     | 589 (9.78%)   | 516 (9.45%)   |
| 6     | 608 (10.09%)  | 545 (9.98%)   |
| 7     | 595 (9.88%)   | 536 (9.81%)   |
| 8     | 655 (10.87%)  | 551 (10.09%)  |
| 9     | 712 (11.82%)  | 550 (10.07%)  |
| 10    | 767 (12.73%)  | 612 (11.2%)   |
| Total | 6024 (100.0%) | 5463 (100.0%) |
+-------+---------------+---------------+
```

[6]:
```
output_obs_table(start_year, end_year, ses_mmb)
```

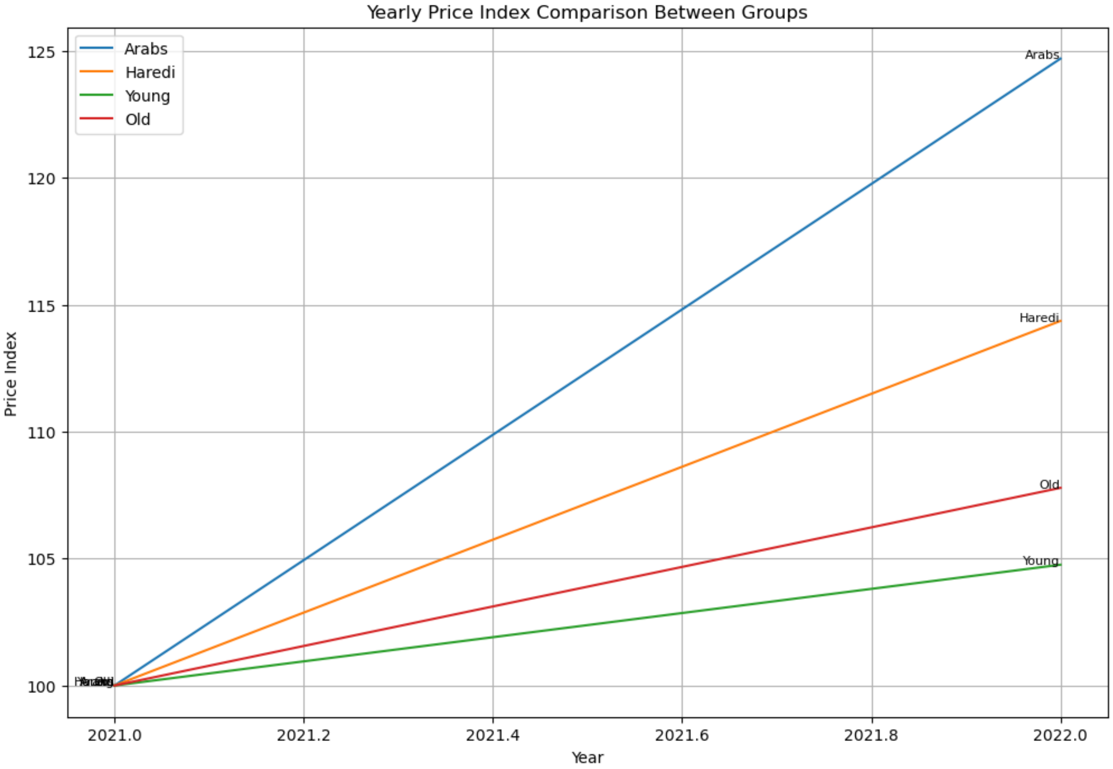
```
+-------+---------------+---------------+
|       | 2021          | 2022          |
|-------+---------------+---------------|
| 1     | 1204 (19.99%) | 1211 (22.17%) |
| 2     | 804 (13.35%)  | 753 (13.78%)  |
| 3     | 1503 (24.95%) | 1352 (24.75%) |
| 4     | 2393 (39.72%) | 2016 (36.9%)  |
| 5     | 153 (2.54%)   | 146 (2.67%)   |
| Total | 6024 (100.0%) | 5463 (100.0%) |
+-------+---------------+---------------+
```

2.2 Plots

2.2.1 Yearly Price Index Comparison Between Groups

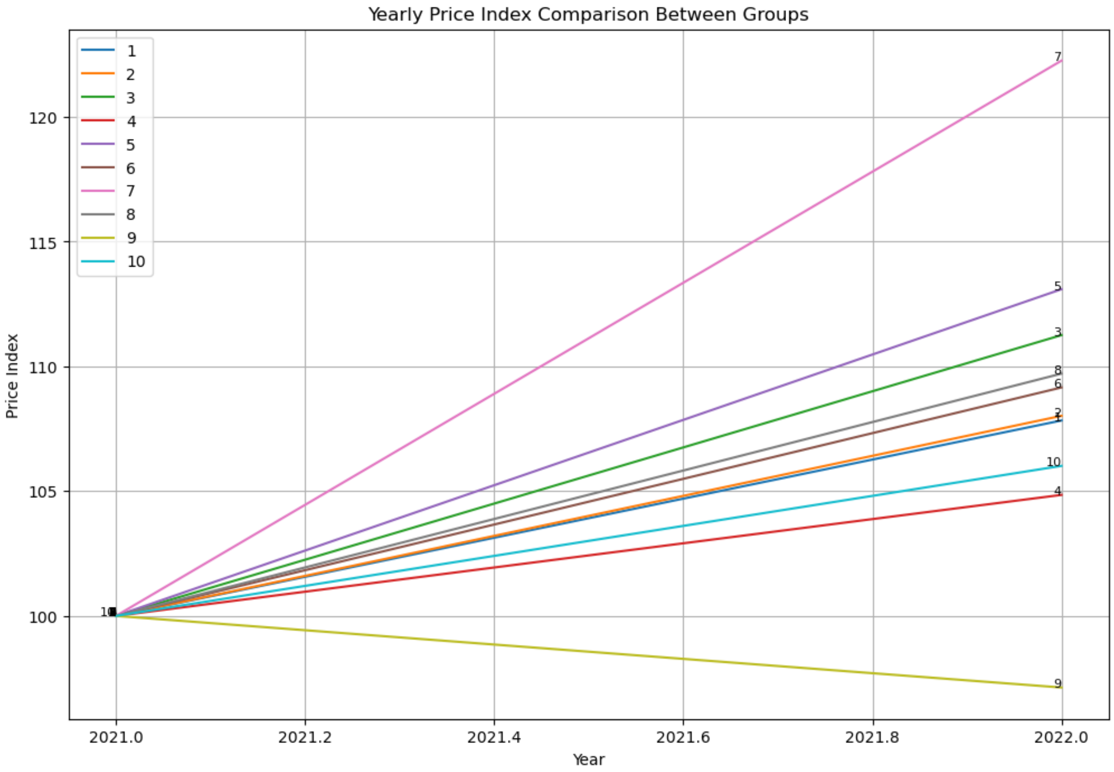
[7]:

```
price_index_over_time(demo_analysis)
```

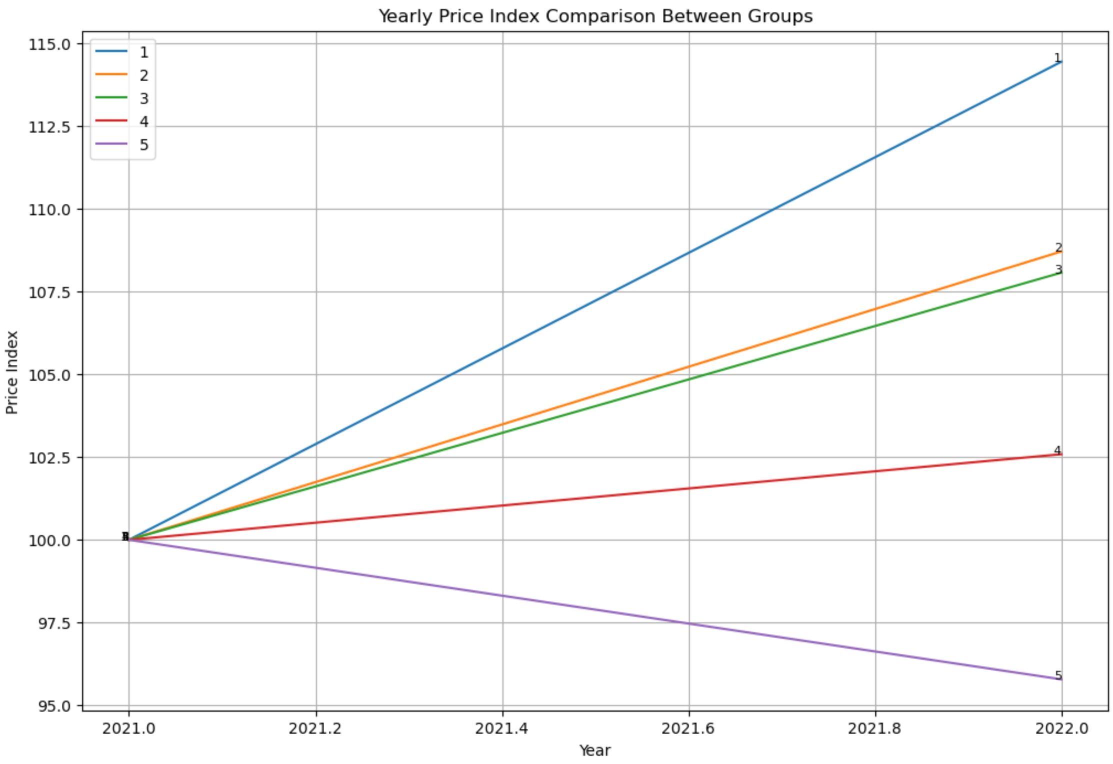


[8]:

```
price_index_over_time(income_analysis)
```



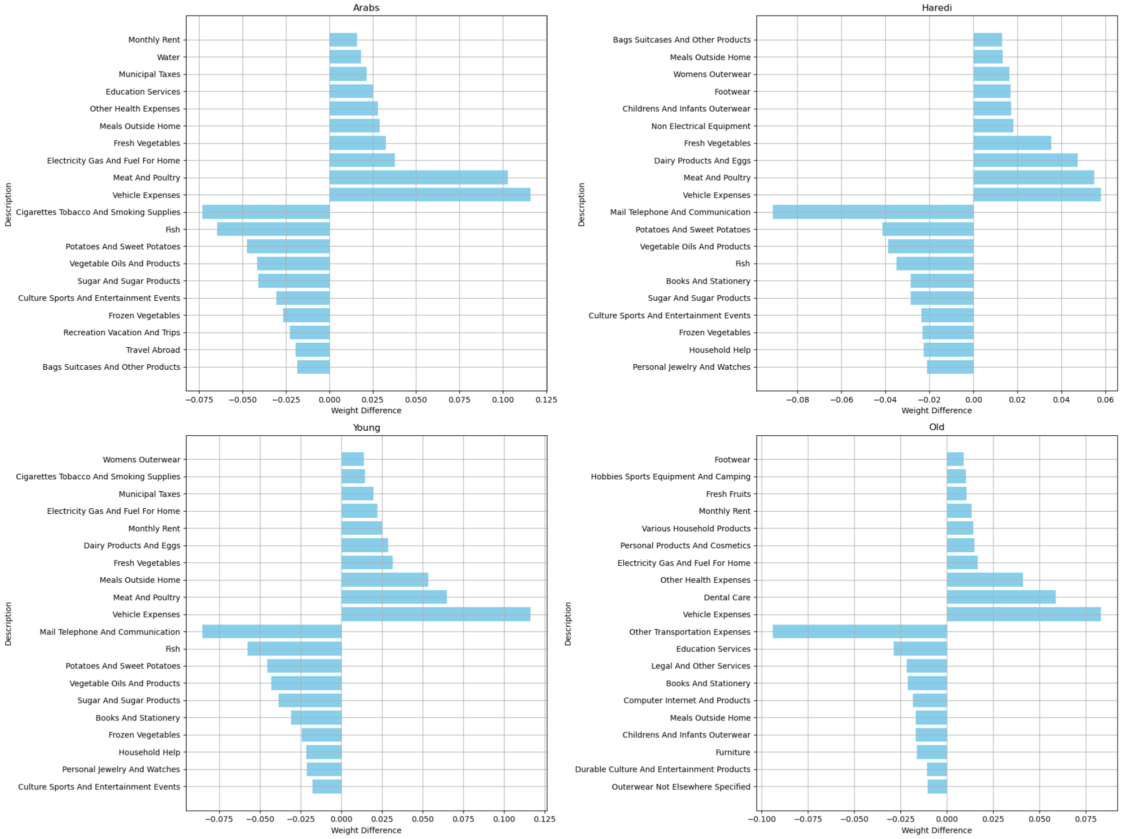
[9]:
```
price_index_over_time(ses_analysis)
```



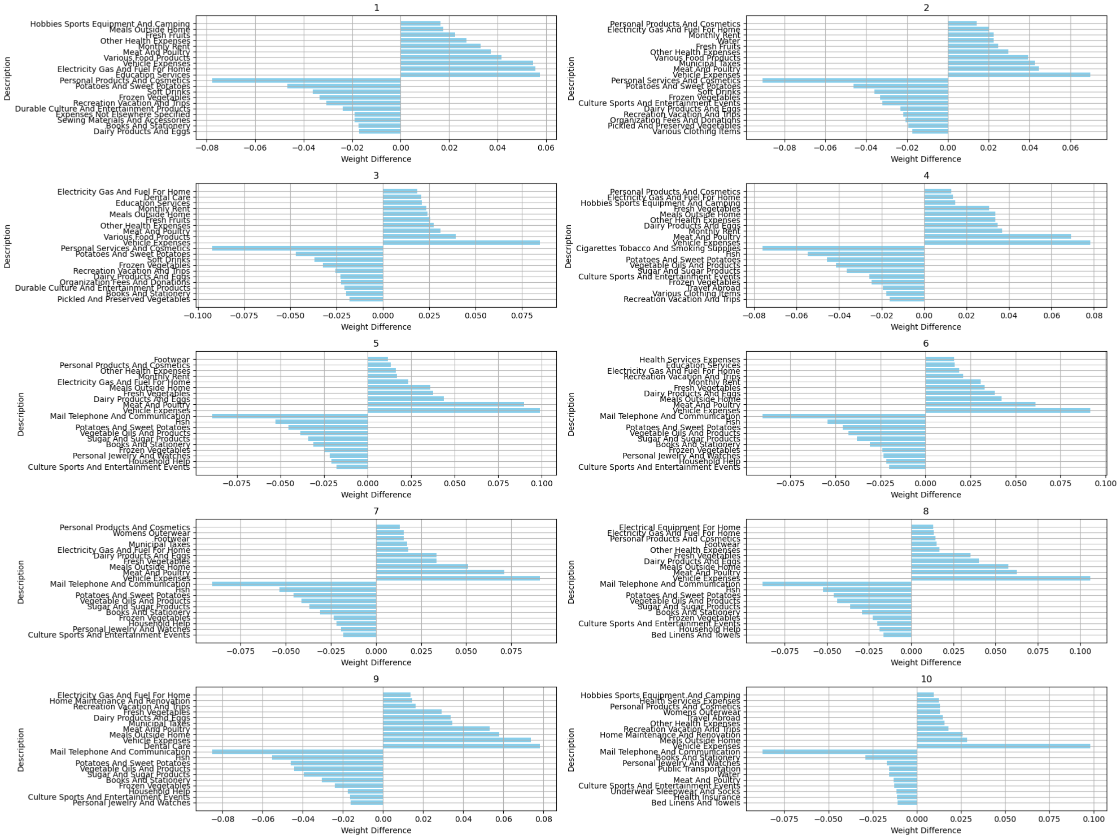
2.2.2 Top Weight Differences

[10]:
```
# Define control group
weights_comparison_control =␣
 ↪gen_pop['combined_secondary_df'][gen_pop['combined_secondary_df']['Year'] ==␣
 ↪comparison_year]
```

[11]:
```
# Top weight differences - demograhic groups
demo_comparison_groups = {}
for group in demo_analysis:
    demo_comparison_groups[group] =␣
 ↪demo_analysis[group]['combined_secondary_df'][demo_analysis[group]['combined_secondary_df']
 ↪== comparison_year]
top_abs_weight_differences(demo_comparison_groups, weights_comparison_control,␣
 ↪top_n)
```

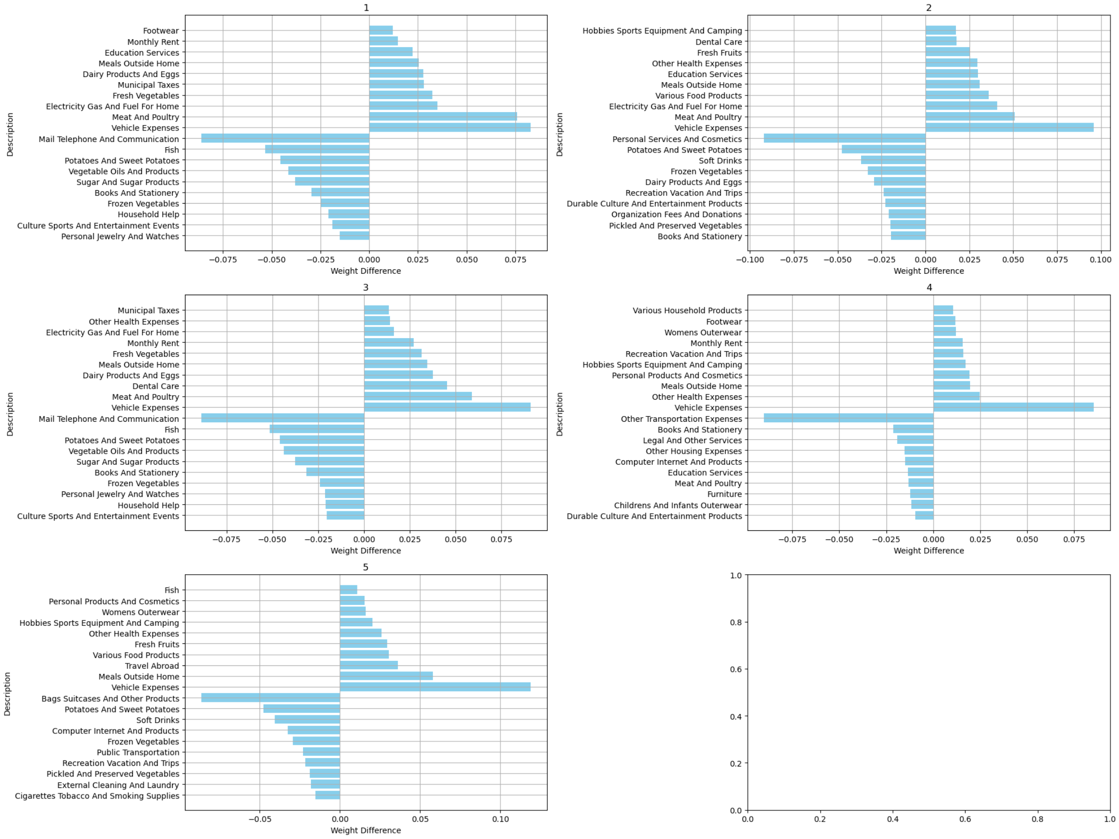


[12]:
```
# Top weight differences - income groups
income_comparison_groups = {}
for group in income_analysis:
    income_comparison_groups[group] =␣
 ↪income_analysis[group]['combined_secondary_df'][income_analysis[group]['combined_secondary_
 ↪== comparison_year]
top_abs_weight_differences(income_comparison_groups,␣
 ↪weights_comparison_control, top_n)
```



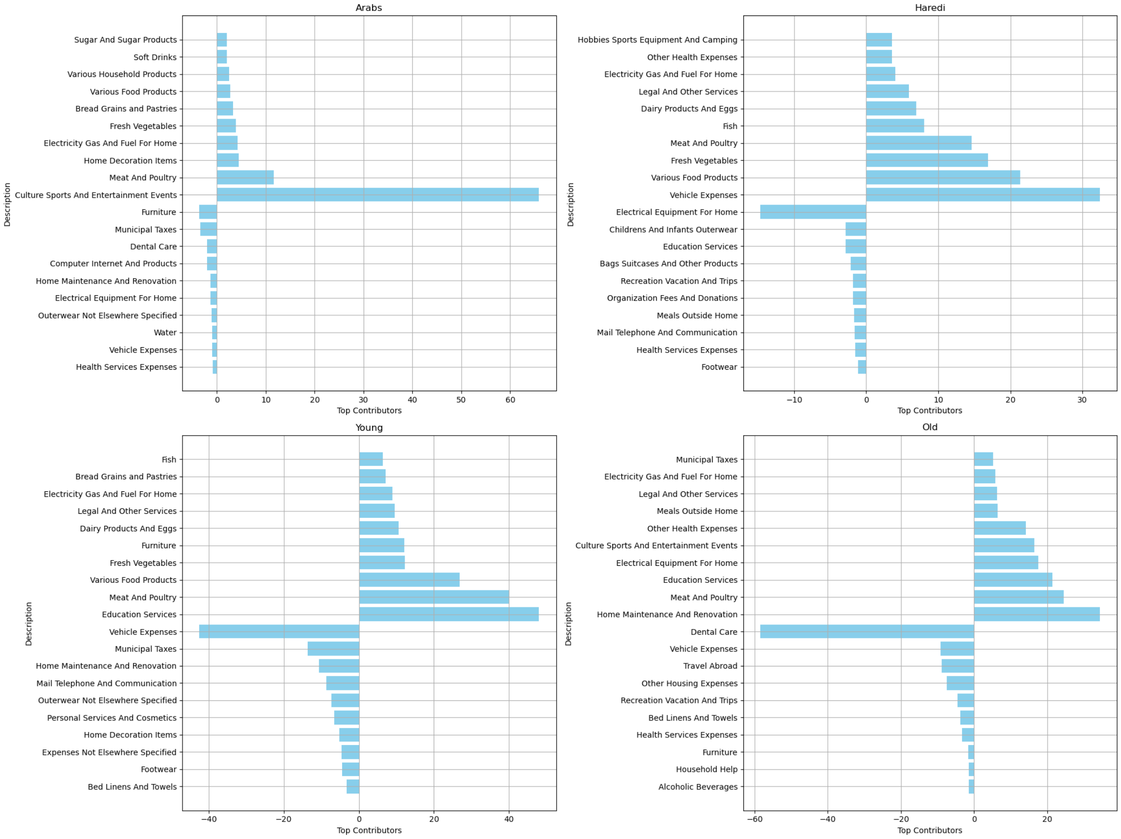
[13]:

```
# Top weight differences - SES groups
ses_comparison_groups = {}
for group in ses_analysis:
    ses_comparison_groups[group] =␣
 ↪ses_analysis[group]['combined_secondary_df'][ses_analysis[group]['combined_secondary_df']['
 ↪== comparison_year]
top_abs_weight_differences(ses_comparison_groups, weights_comparison_control,␣
 ↪top_n)
```

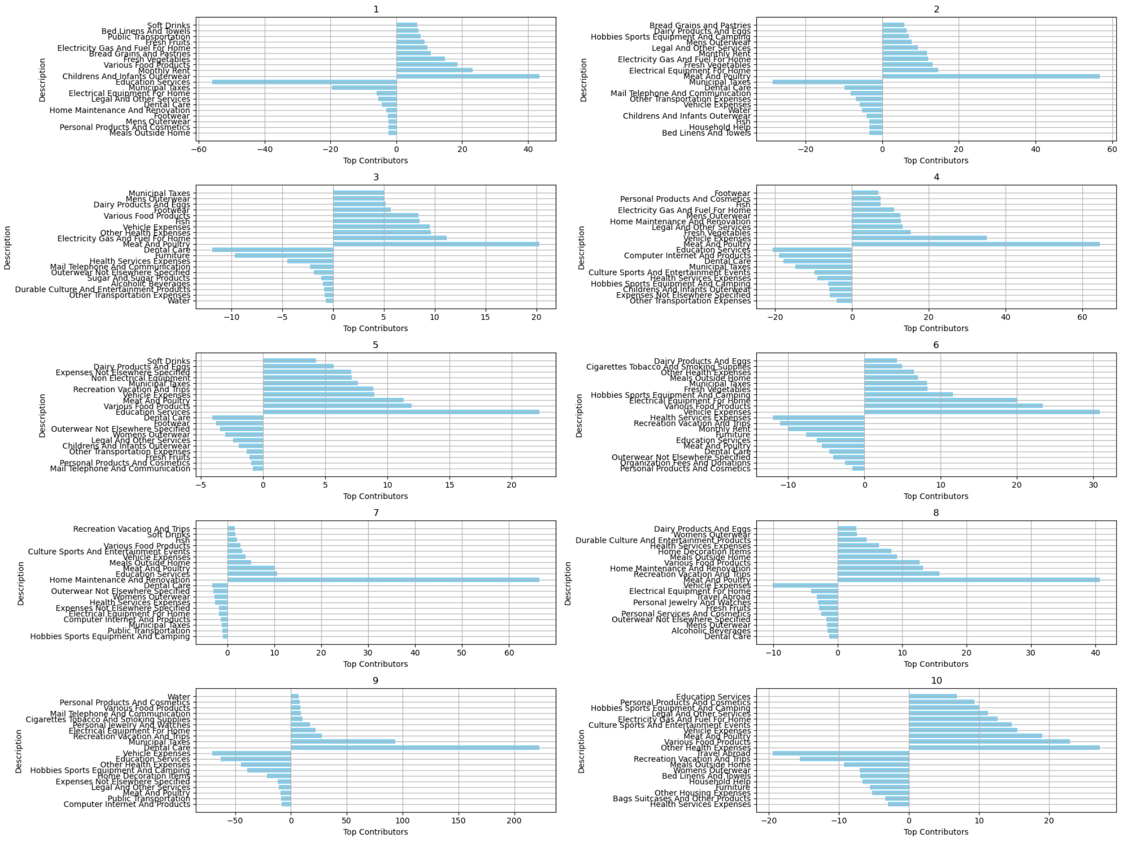


2.2.3 Top Contributors to CPI Change

[14]:
```
# Top contributors - demographic groups
demo_yearly_price_indexes = {}
for group in demo_analysis:
    demo_yearly_price_indexes[group] =␣
 ↪demo_analysis[group]['yearly_price_index'][comparison_year]
top_price_index_contributors(demo_comparison_groups, demo_yearly_price_indexes,␣
 ↪top_n)
```

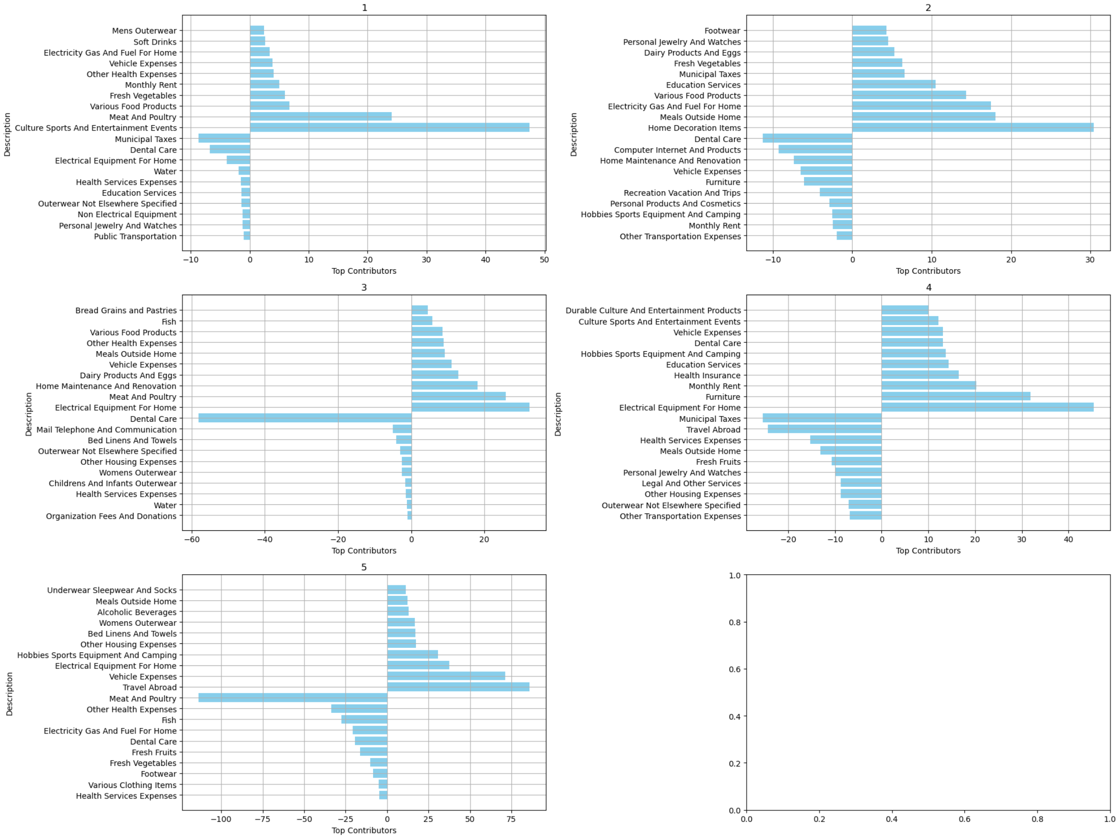


```
[15]: # Top contributors - income groups
      income_yearly_price_indexes = {}
      for group in income_analysis:
          income_yearly_price_indexes[group] =␣
        ↪income_analysis[group]['yearly_price_index'][comparison_year]
      top_price_index_contributors(income_comparison_groups,␣
        ↪income_yearly_price_indexes, top_n)
```



[16]:

```
# Top contributors - SES groups
ses_yearly_price_indexes = {}
for group in ses_analysis:
    ses_yearly_price_indexes[group] =␣
 ↪ses_analysis[group]['yearly_price_index'][comparison_year]
top_price_index_contributors(ses_comparison_groups, ses_yearly_price_indexes,␣
 ↪top_n)
```


1
Mens Outerwear
Soft Drinks
Electricity Gas And Fuel For Home
Vehicle Expenses
Other Health Expenses
Monthly Rent
Fresh Vegetables
Various Food Products
Meat And Poultry
Culture Sports And Entertainment Events
Municipal Taxes
Dental Care
Electrical Equipment For Home
Water
Health Services Expenses
Education Services
Outerwear Not Elsewhere Specified
Non Electrical Equipment
Personal Jewelry And Watches
Public Transportation
Description
−10
0
10
20
30
40
50
Top Contributors
2
Footwear
Personal Jewelry And Watches
Dairy Products And Eggs
Fresh Vegetables
Municipal Taxes
Education Services
Various Food Products
Electricity Gas And Fuel For Home
Meals Outside Home
Home Decoration Items
Dental Care
Computer Internet And Products
Home Maintenance And Renovation
Vehicle Expenses
Furniture
Recreation Vacation And Trips
Personal Products And Cosmetics
Hobbies Sports Equipment And Camping
Monthly Rent
Other Transportation Expenses
Description
−10
0
10
20
30
Top Contributors
3
Bread Grains and Pastries
Fish
Various Food Products
Other Health Expenses
Meals Outside Home
Vehicle Expenses
Dairy Products And Eggs
Home Maintenance And Renovation
Meat And Poultry
Electrical Equipment For Home
Dental Care
Mail Telephone And Communication
Bed Linens And Towels
Outerwear Not Elsewhere Specified
Other Housing Expenses
Womens Outerwear
Childrens And Infants Outerwear
Health Services Expenses
Water
Organization Fees And Donations
Description
−60
−40
−20
0
20
Top Contributors
4
Durable Culture And Entertainment Products
Culture Sports And Entertainment Events
Vehicle Expenses
Dental Care
Hobbies Sports Equipment And Camping
Education Services
Health Insurance
Monthly Rent
Furniture
Electrical Equipment For Home
Municipal Taxes
Travel Abroad
Health Services Expenses
Meals Outside Home
Fresh Fruits
Personal Jewelry And Watches
Legal And Other Services
Other Housing Expenses
Outerwear Not Elsewhere Specified
Other Transportation Expenses
Description
−20
−10
0
10
20
30
40
Top Contributors
5
Underwear Sleepwear And Socks
Meals Outside Home
Alcoholic Beverages
Womens Outerwear
Bed Linens And Towels
Other Housing Expenses
Hobbies Sports Equipment And Camping
Electrical Equipment For Home
Vehicle Expenses
Travel Abroad
Meat And Poultry
Other Health Expenses
Fish
Electricity Gas And Fuel For Home
Dental Care
Fresh Fruits
Fresh Vegetables
Footwear
Various Clothing Items
Health Services Expenses
Description
−100
−75
−50
−25
0
25
50
75
Top Contributors
1.0
0.8
0.6
0.4
0.2
0.0
0.0
0.2
0.4
0.6
0.8
1.0